The Structure of Concurrent Process Histories

Chad Nester*

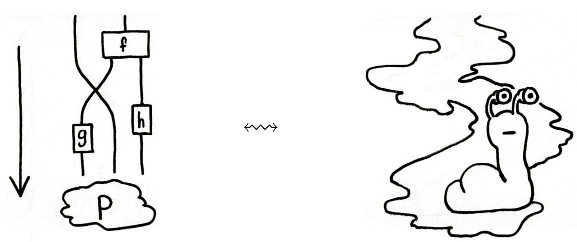
Tallinn University of Technology

Abstract. We identify the algebraic structure of the material histories generated by concurrent processes. Specifically, we extend existing categorical theories of resource convertibility to capture concurrent interaction. Our formalism admits an intuitive graphical presentation via string diagrams for proarrow equipments.

1 Introduction

Concurrent systems are abundant in computing, and indeed in the world at large. Despite the large amount of attention paid to the modelling of concurrency in recent decades (e.g., [1, 10, 16–18]), a canonical mathematical account has yet to emerge, and the basic structure of concurrent systems remains elusive.

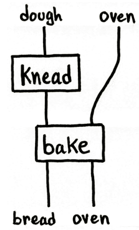
In this paper we present a basic structure that captures what we will call the *material* aspect of concurrent systems: As a process unfolds in time it leaves behind a material history of effects on the world, like the way a slug moving through space leaves a trail of slime. This slime is captured in a natural way by *resource theories* in the sense of [4], in which morphisms of symmetric monoidal categories – conveniently expressed as string diagrams – are understood as transformations of resources.



From the resource theoretic perspective, objects of a symmetric monoidal category are understood as collections of resources, with the unit object denoting the empty collection and the tensor product of two collections consisting of their combined contents. Morphisms are understood as ways to transform one collection of resources into another, which may be combined sequentially via composition, and in parallel via the tensor product. For example, the process of

* This research was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

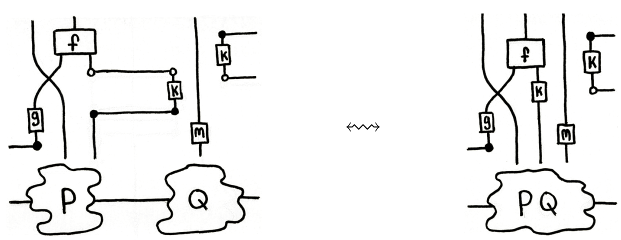
baking bread might generate the following material history:



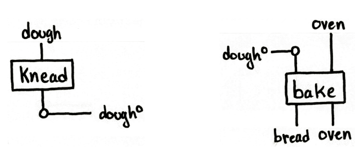
meaning that the baking process involved kneading dough and baking it in an oven to obtain bread (and also the oven).

This approach to expressing the material history of a process has many advantages: It is general, in that it assumes minimal structure; canonical, in that monoidal categories are well-studied as mathematical objects; and relatively friendly, as it admits an intuitive graphical calculus (string diagrams). However, it is unable to capture the interaction between components of a concurrent process. For example, consider our hypothetical baking process and suppose that the kneading and baking of the dough are handled by separate subsystems, with control of the dough being handed to the baking subsystem once the kneading is complete. Such interaction of parts is a fundamental aspect of concurrency, but is not expressible in this framework – we can only describe the effects of the system as a whole.

We remedy this by extending a given resource theory to allow the decomposition of material histories into concurrent components. Specifically, we augment the string diagrams for symmetric monoidal categories with *corners*, through which resources may flow between different components of a transformation.



Returning to our baking example, we might express the material history of the kneading and baking subsystems *separately* with the following diagrams, which may be composed horizontally to obtain the material history of the baking process as a whole.



These augmented diagrams denote cells of a single object double category constructed from the original resource theory. The corners make this double category into a proarrow equipment, which turns out to be all the additional structure we need in order to express concurrent interaction. From only this structure, we obtain a theory of exchanges – a sort of minimal system of behavioural types – that conforms to our intuition about how such things ought to work remarkably well.

Our approach to these concurrent material histories retains the aforementioned advantages of the resource-theoretic perspective: We lose no generality, since our construction applies to any resource theory; It is canonical, with proarrow equipments being a fundamental structure in formal category theory – although not usually seen in such concrete circumstances; Finally, it remains relatively friendly, since the string diagrams for monoidal categories extend in a natural way to string diagrams for proarrow equipments [11].

1.1 Contributions and Related Work

Related Work. Monoidal categories are ubiquitous – if often implicit – in theoretical computer science. An example from the theory of concurrency is [15], in which monoidal categories serve a purpose similar to their purpose here. String diagrams for monoidal categories seem to have been invented independently a number of times, but until recently were uncommon in printed material due to technical limitations. The usual reference is [12]. We credit the resource-theoretic interpretation of monoidal categories and their string diagrams to [4]. Double categories first appear in [6]. Free double categories are considered in [5] and again in [7]. The idea of a proarrow equipment first appears in [22], albeit in a rather different form. Proarrow equipments have subsequently appeared under many names in formal category theory (see e.g., [9, 20]). String diagrams for double categories and proarrow equipments are treated precisely in [11]. We have been inspired by work on message passing and behavioural types, in particular [2], from which we have adopted our notation for exchanges.

Contributions. Our main contribution is the resource-theoretic interpretation of certain proarrow equipments, which we call *cornerings*, and the observation that they capture exactly the structure of concurrent process histories. Our mathematical contributions are minor, most significantly the identification of crossing cells in the free cornering of a resource theory and the corresponding Lemma 2, which we believe to be novel. We do not claim the other lemmas of the paper as significant mathematical contributions. Instead, they serve to flesh out the structure of the free cornering.

1.2 Organization and Prerequisites

Prerequisites. This paper is largely self-contained, but we assume some familiarity with category theory, in particular with monoidal categories and their string diagrams. Some good references are [8, 14, 19].

Organization. In Section 2 we review the resource-theoretic interpretation of symmetric monoidal categories. We continue by reviewing the theory of double categories in Section 3, specialized to the single object case. In Section 4 we introduce cornerings of a resource theory, in particular the free such cornering, and exhibit the existence of crossing cells in the free cornering. In Section 5 we show how the free cornering of a resource theory inherits its resource-theoretic interpretation while enabling the concurrent decomposition of resource transformations. In Section 6 we conclude and consider directions for future work.

2 Monoidal Categories as Resource Theories

Symmetric strict monoidal categories can be understood as theories of resource transformation. Objects are interpreted as collections of resources, with $A \otimes B$ the collection consisting of both A and B, and I the empty collection. Arrows $f : A \to B$ are understood as ways to transform the resources of A into those of B. We call symmetric strict monoidal categories *resource theories* when we have this sort of interpretation in mind.

For example, let $\mathfrak{B}$ be the free symmetric strict monoidal category with generating objects

$$\{\mathtt{bread}, \mathtt{dough}, \mathtt{water}, \mathtt{flour}, \mathtt{oven}\}$$

and with generating arrows

$$\mathtt{mix} : \mathtt{water} \otimes \mathtt{flour} \to \mathtt{dough} \qquad \mathtt{knead} : \mathtt{dough} \to \mathtt{dough}$$

$$\mathtt{bake} : \mathtt{dough} \otimes \mathtt{oven} \to \mathtt{bread} \otimes \mathtt{oven}$$

subject to no equations. $\mathfrak{B}$ can be understood as a resource theory of baking bread. The arrow `mix` represents the process of combining water and flour to form a bread dough, `knead` represents kneading dough, and `bake` represents baking dough in an oven to obtain bread (and an oven).

The structure of symmetric strict monoidal categories provides natural algebraic scaffolding for composite transformations. For example, consider the following arrow of $\mathfrak{B}$:

$$(\mathtt{bake} \otimes 1_{\mathtt{dough}}); (1_{\mathtt{bread}} \otimes \sigma_{\mathtt{oven},\mathtt{dough}}; \mathtt{bake})$$

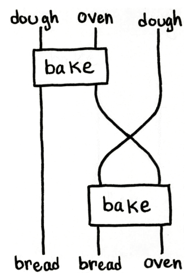
of type

$$\mathtt{dough} \otimes \mathtt{oven} \otimes \mathtt{dough} \to \mathtt{bread} \otimes \mathtt{bread} \otimes \mathtt{oven}$$

where $\sigma_{A,B} : A \otimes B \xrightarrow{\sim} B \otimes A$ is the braiding. This arrow describes the transformation of two units of dough into loaves of bread by baking them one after the other in an oven.

It is often more intuitive to write composite arrows like this as string diagrams: Objects are depicted as wires, and arrows as boxes with inputs and outputs. Composition is represented by connecting output wires to input wires, and we represent the tensor product of two morphisms by placing them beside one another. Finally, the braiding is represented by crossing the wires involved.

For the morphism discussed above, the corresponding string diagram is:

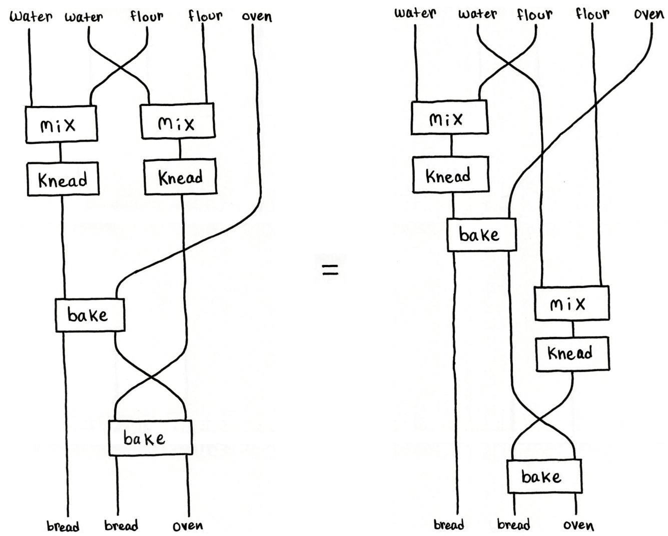


Notice how the topology of the diagram captures the logical flow of resources.

Given a pair of parallel arrows $f, g : A \to B$ in some resource theory, both f and g are ways to obtain B from A, but they may not have the same effect on the resources involved. We explain by example: Consider the parallel arrows $1_{\texttt{dough}}, \texttt{knead} : \texttt{dough} \to \texttt{dough}$ of $\mathfrak{B}$. Clearly these should not be understood to have the same effect on the dough in question, and this is reflected in $\mathfrak{B}$ by the fact that they are not made equal by its axioms. Similarly, `knead` and $\texttt{knead} \circ \texttt{knead}$ are not equal in $\mathfrak{B}$, which we understand to mean that kneading dough twice does not have the same effect as kneading it once, and that in turn any `bread` produced from twice-kneaded dough will be different from once-kneaded bread in our model.

Consider a hypothetical resource theory constructed from $\mathfrak{B}$ by imposing the equation $\texttt{knead} \circ \texttt{knead} = \texttt{knead}$. In this new setting we understand kneading dough once to have the same effect as kneading it twice, three times, and so on, because the corresponding arrows are all equal. Of course, the sequence of events described by `knead` is not the one described by $\texttt{knead} \circ \texttt{knead}$: In the former the dough has been kneaded only once, while in the latter it has been kneaded twice. The equality of the two arrows indicates that these two different processes would have the same effect on the dough involved. We adopt as a general principle in our design and understanding of resource theories that transformations should be equal if and only if they have the same effect on the resources involved.

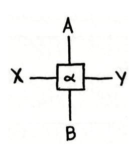
For the sake of further illustration, observe that by naturality of the braiding maps the following two resource transformations are equal in $\mathfrak{B}$:



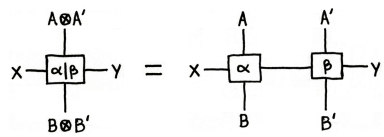
Each transformation gives a method of baking two loaves of bread. On the left, two batches of dough are mixed and kneaded before being baked one after the other. On the right, first one batch of dough is mixed, kneaded and baked and only then is the second batch mixed, kneaded, and baked. Their equality tells us that, according to $\mathfrak{B}$, the two procedures will have the same effect, resulting in the same bread when applied to the same ingredients with the same oven.

3 Single Object Double Categories

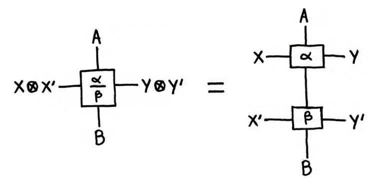
In this section we set up the rest of our development by presenting the theory of *single object double categories*, being those double categories $\mathbb{D}$ with exactly one object. In this case $\mathbb{D}$ consists of a *horizontal edge monoid* $\mathbb{D}_H = (\mathbb{D}_H, \otimes, I)$, a *vertical edge monoid* $\mathbb{D}_V = (\mathbb{D}_V, \otimes, I)$, and a collection of *cells*



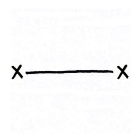
where $A, B \in \mathbb{D}_H$ and $X, Y \in \mathbb{D}_V$. Given cells α, β where the right boundary of α matches the left boundary of β we may form a cell $\alpha|\beta$ – their *horizontal composite* – and similarly if the bottom boundary of α matches the top boundary of β we may form $\frac{\alpha}{\beta}$ – their *vertical composite* – with the boundaries of the composite cell formed from those of the component cells using $\otimes$. We depict horizontal and vertical composition, respectively, as in:



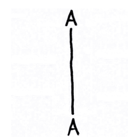
and



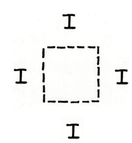
Horizontal and vertical composition of cells are required to be associative and unital. We omit wires of sort I in our depictions of cells, allowing us to draw horizontal and vertical identity cells, respectively, as in:

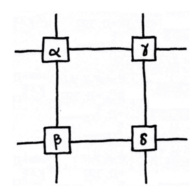


and

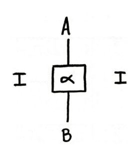


Finally, the horizontal and vertical identity cells of type I must coincide – we write this cell as $\square_I$ and depict it as empty space, see below on the left – and vertical and horizontal composition must satisfy the interchange law. That is, $\frac{\alpha}{\beta}|\frac{\gamma}{\delta} = \frac{\alpha|\gamma}{\beta|\delta}$, allowing us to unambiguously interpret the diagram below on the right:

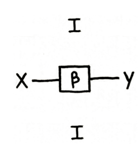




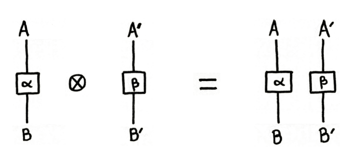
Every single object double category $\mathbb{D}$ defines strict monoidal categories $\mathbf{V}\mathbb{D}$ and $\mathbf{H}\mathbb{D}$, consisting of the cells for which the $\mathbb{D}_H$ and $\mathbb{D}_V$ valued boundaries respectively are all I, as in:



and



That is, the collection of objects of $\mathbf{V}\mathbb{D}$ is $\mathbb{D}_H$, composition in $\mathbf{V}\mathbb{D}$ is vertical composition of cells, and the tensor product in $\mathbf{V}\mathbb{D}$ is given by horizontal composition:



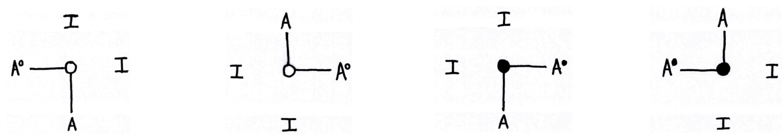
In this way, $\mathbf{V}\mathbb{D}$ forms a strict monoidal category, which we call the category of *vertical cells* of $\mathbb{D}$. Similarly, $\mathbf{H}\mathbb{D}$ is also a strict monoidal category (with collection of objects $\mathbb{D}_V$) which we call the *horizontal cells* of $\mathbb{D}$.

4 Cornerings and Crossings

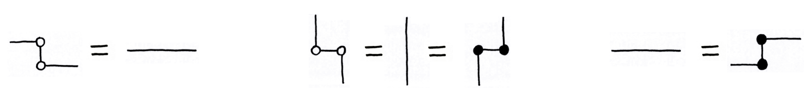
Next, we define cornerings, our primary technical device. In particular we discuss the free cornering of a resource theory, which we show contains special crossing cells with nice formal properties. Tersely, a cornering of a resource theory $\mathbb{A}$ is a single object proarrow equipment with $\mathbb{A}$ as its vertical cells. Explicitly:

Definition 1. *Let $\mathbb{A}$ be a symmetric strict monoidal category. Then a* cornering *of $\mathbb{A}$ is a single object double category $\mathbb{D}$ such that:*

(i) The vertical cells of $\mathbb{D}$ are $\mathbb{A}$. That is, there is an isomorphism of categories $\mathbf{V}\mathbb{D} \cong \mathbb{A}$.
(ii) For each A in $\mathbb{A}_0 \cong \mathbb{D}_H$, there are distinguished elements A° and $A^\bullet$ of $\mathbb{D}_V$ along with distinguished cells of $\mathbb{D}$



called ∘-corners *and* •-corners *respectively, which must satisfy the* yanking equations*:*

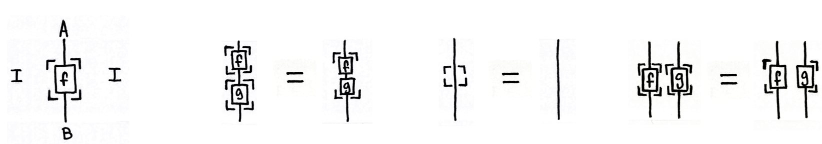


Intuitively, A° denotes an instance of A moving from left to right, and $A^\bullet$ denotes an instance of A moving from right to left (see Section 5).

Of particular interest is the free cornering of a resource theory:

Definition 2. *Let $\mathbb{A}$ be a resource theory. Then the* free cornering of $\mathbb{A}$, *written $\ulcorner \mathbb{A} \urcorner$, is the free single object double category defined as follows:*

- *The horizontal edge monoid $\ulcorner \mathbb{A} \urcorner_H = (\mathbb{A}_0, \otimes, I)$ is given by the objects of $\mathbb{A}$.*
- *The vertical edge monoid $\ulcorner \mathbb{A} \urcorner_V = (\mathbb{A}_0 \times \{\circ, \bullet\})^*$ is the free monoid on the set $\mathbb{A}_0 \times \{\circ, \bullet\}$ of polarized objects of $\mathbb{A}$ – whose elements we write A° and $A^\bullet$.*
- *The generating cells consist of corners for each object A of $\mathbb{A}$ as above, subject to the yanking equations, along with a vertical cell $\ulcorner f \urcorner$ for each morphism $f : A \to B$ of $\mathbb{A}$ subject to equations as in:*

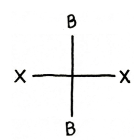


For a precise development of free double categories see [7]. In brief: cells are formed from the generating cells by horizontal and vertical composition, subject to the axioms of a double category in addition to any generating equations. The free cornering is free both in the sense that it is freely generated, and in the sense that for any cornering $\mathbb{D}$ of $\mathbb{A}$ there is exactly one double functor $\ulcorner\mathbb{A}\lrcorner \to \mathbb{D}$ that sends corner cells to corner cells and restricts to the identity on $\mathbb{A} \cong \mathbf{V}\mathbb{D}$. That is, diagrams in $\ulcorner\mathbb{A}\lrcorner$ have a canonical interpretation in any cornering of $\mathbb{A}$.

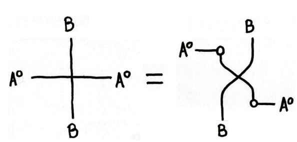
Proposition 1. *$\ulcorner\mathbb{A}\lrcorner$ is a cornering of $\mathbb{A}$.*

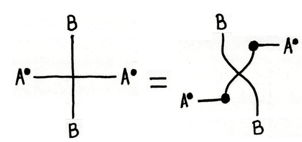
Proof. Intuitively $\mathbf{V}\ulcorner\mathbb{A}\lrcorner \cong \mathbb{A}$ because in a composite vertical cell every wire bent by a corner must eventually be un-bent by the matching corner, which by yanking is the identity. The only other generators are the cells $\ulcorner f \lrcorner$, and so any vertical cell in $\ulcorner A \lrcorner$ can be written as $\ulcorner g \lrcorner$ for some morphism g of $\mathbb{A}$. A more rigorous treatment of corner cells can be found in [11], to the same effect. □

Before we properly explain our interest in $\ulcorner\mathbb{A}\lrcorner$ we develop a convenient bit of structure: *crossing cells.* For each B of $\ulcorner\mathbb{A}\lrcorner_H$ and each X of $\ulcorner\mathbb{A}\lrcorner_V$ we define a cell

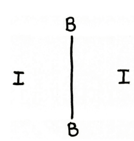


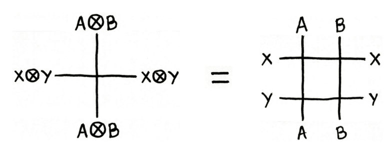
of $\ulcorner\mathbb{A}\lrcorner$ inductively as follows: In the case where X is A° or $A^\bullet$, respectively, define the crossing cell as in the diagrams below on the left and right, respectively:





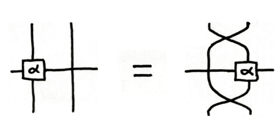
in the case where X is I, define the crossing cell as in the diagram below on the left, and in the composite case define the crossing cell as in the diagram below on the right:



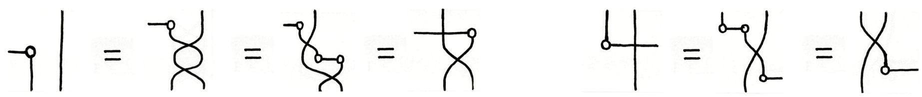


We prove a technical lemma:

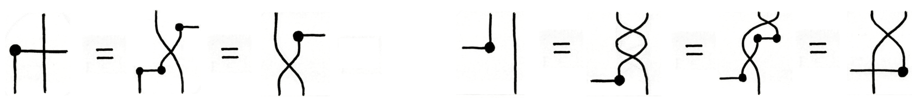
Lemma 1. *For any cell α of $\ulcorner\mathbb{A}\lrcorner$ we have*



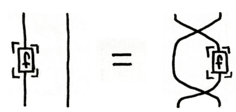
Proof. By structural induction on cells of $\ulcorner\mathbb{A}\urcorner$. For the $\circ$-corners we have:



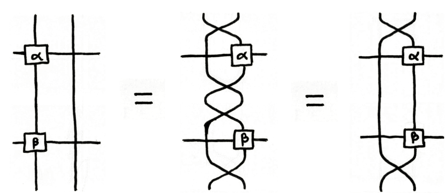
and for the $\bullet$-corners, similarly:



the final base cases are the $\ulcorner f \urcorner$ maps:



There are two inductive cases. For vertical composition, we have:

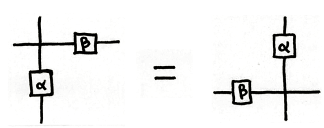


Horizontal composition is similarly straightforward, and the claim follows by induction.

□

From this we obtain a "non-interaction" property of our crossing cells, similar to the naturality of braiding in symmetric monoidal categories:

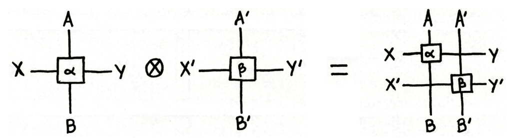
Corollary 1. *For cells α of $\mathbf{V}\ulcorner\mathbb{A}\urcorner$ and β of $\mathbf{H}\ulcorner\mathbb{A}\urcorner$, the following equation holds in $\ulcorner\mathbb{A}\urcorner$:*



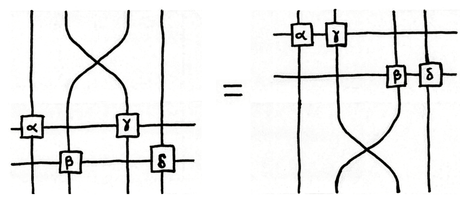
These crossing cells greatly aid in the legibility of diagrams corresponding to cells in $\ulcorner\mathbb{A}\urcorner$, but also tell us something about the categorical structure of $\ulcorner\mathbb{A}\urcorner$, namely that it is a monoidal double category in the sense of [21]:

Lemma 2. *If $\mathbb{A}$ is a symmetric strict monoidal category then $\ulcorner\mathbb{A}\urcorner$ is a monoidal double category. That is, $\ulcorner\mathbb{A}\urcorner$ is a pseudo-monoid object in the strict 2-category* **V**DblCat *of double categories, lax double functors, and vertical transformations.*

Proof. We give the action of the tensor product on cells:



This defines a pseudofunctor, with the component of the required vertical transformation given by exchanging the two middle wires as in:

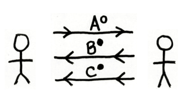


Notice that $\otimes$ is strictly associative and unital, in spite of being only pseudofunctorial.

□

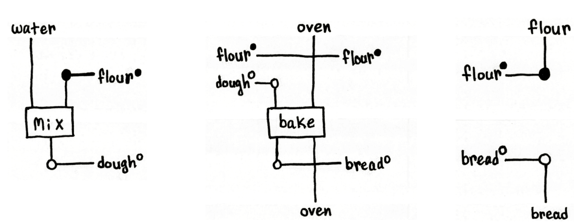
5 Concurrency Through Cornering

We next proceed to extend the resource-theoretic interpretation of some symmetric strict monoidal category $\mathbb{A}$ to its free cornering ${}^{\ulcorner}_{\llcorner}\mathbb{A}^{\urcorner}_{\lrcorner}$. Interpret elements of ${}^{\ulcorner}_{\llcorner}\mathbb{A}^{\urcorner}_{\lrcorner V}$ as $\mathbb{A}$-*valued exchanges*. Each exchange $X_1 \otimes \cdots \otimes X_n$ involves a left participant and a right participant giving each other resources in sequence, with A° indicating that the left participant should give the right participant an instance of A, and $A^\bullet$ indicating the opposite. For example say the left participant is `Alice` and the right participant is `Bob`. Then we can picture the exchange $A^\circ \otimes B^\bullet \otimes C^\bullet$ as:

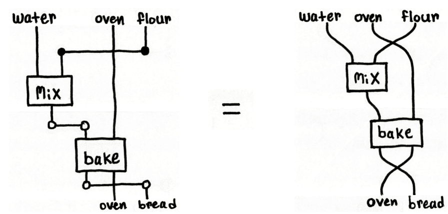
`Alice` $\rightsquigarrow$  $\leftsquigarrow$ `Bob`

Think of these exchanges as happening *in order*. For example the exchange pictured above demands that first `Alice` gives `Bob` an instance of A, then `Bob` gives `Alice` an instance of B, and then finally `Bob` gives `Alice` an instance of C.

We interpret cells of ${}^{\ulcorner}_{\llcorner}\mathbb{A}^{\urcorner}_{\lrcorner}$ as *concurrent transformations*. Each cell describes a way to transform the collection of resources given by the top boundary into that given by the bottom boundary, via participating in $\mathbb{A}$-valued exchanges along the left and right boundaries. For example, consider the following cells of ${}^{\ulcorner}_{\llcorner}\mathfrak{B}^{\urcorner}_{\lrcorner}$:



From left to right, these describe: A procedure for transforming `water` into nothing by `mix`ing it with `flour` obtained by exchange along the right boundary, then sending the resulting `dough` away along the right boundary; A procedure for transforming an `oven` into an `oven`, receiving `flour` along the right boundary and sending it out the left boundary, then receiving `dough` along the left boundary, which is `bake`d in the `oven`, with the resulting `bread` sent out along the right boundary; Finally, a procedure for turning `flour` into `bread` by giving it away and then receiving `bread` along the left boundary. When we compose these concurrent transformations horizontally in the evident way, they give a transformation of resources in the usual sense, i.e., a morphism of $\mathbb{A} \cong \mathbf{V}\ulcorner\mathbb{A}\urcorner$:



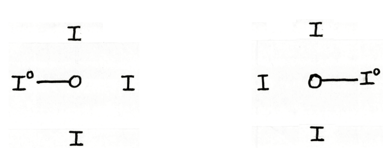
We understand equality of cells in $\ulcorner\mathbb{A}\urcorner$ much as we understand equality of morphisms in a resource theory: two cells should be equal in case the transformations they describe would have the same effect on the resources involved. In this way, cells of $\ulcorner\mathbb{A}\urcorner$ allow us to break a transformation into many concurrent parts. Note that with the crossing cells, it is possible to exchange resources "across" cells.

Consider the category $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ of horizontal cells. If the vertical cells $\mathbf{V}\ulcorner\mathbb{A}\urcorner$ are concerned entirely with the transformation of resources, then our interpretation tells us that the horizontal cells are concerned entirely with exchange. Just as isomorphic objects in $\mathbf{V}\ulcorner\mathbb{A}\urcorner \cong \mathbb{A}$ can be thought of as equivalent collections of resources – being freely transformable into each other – we understand isomorphic objects in $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ as *equivalent exchanges*. For example, There are many ways for `Alice` to give `Bob` an A and a B: Simultaneously, as $A \otimes B$; one after the other, as A and then B; or in the other order, as B and then A. While these are different sequences of events, they achieve the same thing, and are thus equivalent. Similarly, for `Alice` to give `Bob` an instance of I is equivalent to nobody doing anything. Formally, we have:

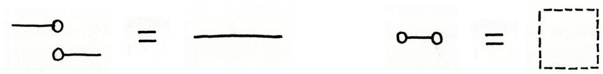
Lemma 3. *In $\mathbf{H}\ulcorner\mathbb{A}\urcorner$ we have for any A, B of $\mathbb{A}$:*

(i) $I^\circ \cong I \cong I^\bullet$.
(ii) $A^\circ \otimes B^\circ \cong B^\circ \otimes A^\circ$ *and* $A^\bullet \otimes B^\bullet \cong B^\bullet \otimes A^\bullet$.
(iii) $(A \otimes B)^\circ \cong A^\circ \otimes B^\circ$ *and* $(A \otimes B)^\bullet \cong A^\bullet \otimes B^\bullet$

Proof. (i) For $I \cong I^\circ$, consider the $\circ$-corners corresponding to I:

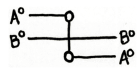


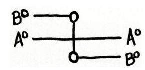
we know that these satisfy the yanking equations:



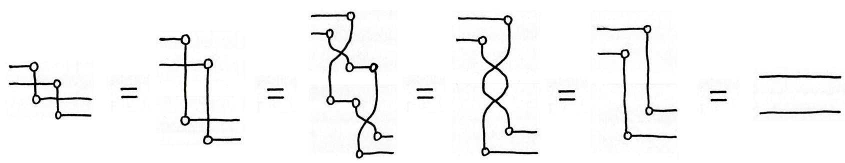
which exhibits an isomorphism $I \cong I^\circ$. Similarly, $I \cong I^\bullet$. Thus, we see formally that exchanging nothing is the same as doing nothing.

(ii) The $\circ$-corner case is the interesting one: Define the components of our isomorphism to be:



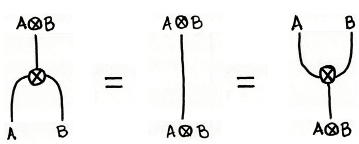
and


then for both of the required composites we have:

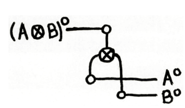


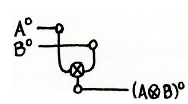
and so $A^\circ \otimes B^\circ \cong B^\circ \otimes A^\circ$. Similarly $A^\bullet \otimes B^\bullet \cong B^\bullet \otimes A^\bullet$. This captures formally the fact that if `Alice` is going to give `Bob` an A and a B, it doesn't really matter which order she does it in.

(iii) Here it is convenient to switch between depicting a single wire of sort $A \otimes B$ and two wires of sort A and B respectively in our string diagrams. To this end, we allow ourselves to depict the identity on $A \otimes B$ in multiple ways, using the notation of [3]:



Then the components of our isomorphism $(A \otimes B)^\circ \cong A^\circ \otimes B^\circ$ are:

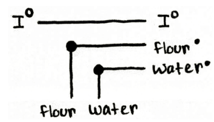


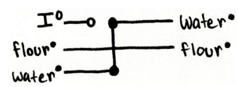
and


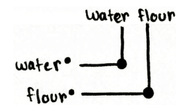
and, much as in (ii), it is easy to see that the two possible composites are both identity maps. Similarly, $(A \otimes B)^\bullet \cong (A^\bullet \otimes B^\bullet)$. This captures formally the fact that giving away a collection is the same thing as giving away its components.

□

For example, we should be able to compose the cells on the left and right below horizontally, since their right and left boundaries, respectively, indicate equivalent exchanges:



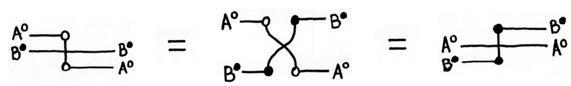




Our lemma tells us that there will always be a canonical isomorphism, as above in the middle, making composition possible.

It is worth noting that we *do not* have $A^\circ \otimes B^\bullet \cong B^\bullet \otimes A^\circ$:

Observation 1 *There is a morphism* $d^{\circ}_{\bullet} : A^\circ \otimes B^\bullet \to B^\bullet \otimes A^\circ$ *in one direction, defined by*



but there is need not be a morphism in the other direction, and this is not in general invertible. In particular, $\mathbf{H}\ulcorner\mathbb{A}\lrcorner$ *is monoidal, but need not be symmetric.*

This observation reflects formally the intuition that if I receive some resources before I am required to send any, then I can send some of the resources that I receive. However, if I must send the resources first, this is not the case. In this way, $\mathbf{H}\ulcorner\mathbb{A}\lrcorner$ contains a sort of causal structure.

6 Conclusions and Future Work

We have shown how to decompose the material history of a process into concurrent components by working in the free cornering of an appropriate resource theory. We have explored the structure of the free cornering in light of this interpretation and found that it is consistent with our intuition about how this sort of thing ought to work. We do not claim to have solved all problems in the modelling of concurrency, but we feel that our formalism captures the material aspect of concurrent systems very well.

We find it quite surprising that the structure required to model concurrent resource transformations is precisely the structure of a proarrow equipment. This structure is already known to be important in formal category theory, and we are appropriately intrigued by its apparent relevance to models of concurrency – a far more concrete setting than the usual context in which one encounters proarrow equipments!

There are of course many directions for future work. For one, our work is inspired by the message passing logic of [2], which has its categorical semantics in *linear actegories*. Any cornering defines an actegory – although not quite a *linear* actegory – and we speculate that cornerings are equivalent to some class of actegories, which would connect our work to the literature on behavioural types. Another direction for future work is to connect our material histories to a theory of concurrent processes – the slugs to our slime – with the goal of a formalism accounting for both. The category of spans of reflexive graphs, interpreted as open transition systems, seems especially promising here [13]. More generally, we would like to know how the perspective presented here can be integrated into other approaches to modelling concurrent systems.

References

1. Samson Abramsky. What are the fundamental structures of concurrency? we still don't know! *CoRR*, abs/1401.4973, 2014.
2. J.R.B. Cockett and C. Pastro. The logic of message-passing. *Science of Computer Programming*, 74:498–533, 2009.
3. J.R.B. Cockett and R.A.G. Seely. Proof theory of the cut rule. In E. Landry, editor, *Categories for the Working Philosopher*, pages 223–261. Oxford University Press, 2017.
4. B. Coecke, T. Fritz, and R.W. Spekkens. A mathematical theory of resources. *Information and Computation*, 250:59–86, 2016.
5. Robert Dawson and Robert Paré. What is a free double category like? *Journal of Pure and Applied Algebra*, 168(1):19–34, 2002.
6. Charles Ehresmann. Catégories structurées. *Annales scientifiques de l'École Normale Supérieure*, 80(4):349–426, 1963.
7. M. Fiore, S. Paoli, and D. Pronk. Model structures on the category of small double categories. *Algebraic and Geometric Topology*, 8(4):1855–1959, 2008.
8. Brendan Fong and David I Spivak. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. 2018.
9. Marco Grandis and Robert Pare. Adjoint for double categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 45(3):193–240, 2004.
10. C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.
11. David Jaz Myers. String Diagrams For Double Categories and Equipments. *arXiv e-prints*, 2016.
12. André Joyal and Ross Street. The geometry of tensor calculus, i. *Advances in Mathematics*, 88(1):55 – 112, 1991.
13. P. Katis, N. Sabadini, and R.F.C Walters. Span(graph): A categorical algebra of transition systems. In *Intnational Conference on Algebraic Methodology and Software Technology*, pages 307–321. Springer, Berlin, Heidelberg, 1997.
14. S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.
15. José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and Computation*, 88(2):105–155, 1990.
16. R. Milner. A calculus of communicating systems. volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.
17. Robin Milner. *Communicating and Mobile Systems: The Pi-Calculus*. Cambridge University Press, USA, 1999.
18. C. A. Petri. Communication with automata. 1966.
19. Peter Selinger. A survey of graphical languages for monoidal categories. In *New Structures for Physics*, pages 289–355. Springer, 2010.
20. Michael Shulman. Framed bicategories and monoidal fibrations. *Theory and Applications of Categories*, 20(18):650–738, 2008.
21. Michael A. Shulman. Constructing symmetric monoidal bicategories. *arXiv e-prints*, 2010.
22. R. J. Wood. Abstract pro arrows i. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 23(3):279–290, 1982.